

Máquinas de Post: Uma Breve Introdução

Luiza Engler Stadelhofer^a, Caroline Sala de Borba^a

^a*Universidade do Estado de Santa Catarina - UDESC*

Centro de Ciências Tecnológicas - CCT

Joinville, Santa Catarina

Abstract

Em 1936, mesmo ano que Turing introduziu ao mundo a sua máquina de Turing, Emil Leon Post desenvolveu a máquina de Post, em busca de criar um modelo de máquina universal o qual teria a habilidade de providenciar uma definição precisa para o conceito do termo algoritmo. O artigo em questão consiste em uma breve introdução à máquina de Post. Nele, apresenta-se seu histórico; sua definição, com o auxílio de um exemplo explicativo; uma visão detalhada de uma de suas aplicações; uma breve discussão sobre sua importância, além da apresentação de um simulador de máquinas de Post desenvolvido pelas autoras.

Keywords: Emil Post, Máquina Universal, Variante da Máquina de Turing

1. Introdução

O termo “algoritmo”, que data até a Idade Média, pode ser definido como um procedimento composto por uma lista finita de instruções [1]. Entretanto, por muito tempo foi tentado criar uma definição formal matemática para este conceito – uma tarefa complexa, se levarmos em consideração o sentido bastante intuitivo que a expressão possui [1].

Contudo, em 1936, mesmo ano que Turing introduziu ao mundo a sua máquina de Turing, Emil Leon Post (1897-1954) desenvolveu a máquina de Post, em busca de criar uma máquina universal capaz de executar qualquer tarefa que possa ser representada por um algoritmo [2]. Modelo o qual teria a habilidade de providenciar uma definição precisa para o conceito do termo algoritmo [2].

O artigo em questão consiste em uma breve introdução à máquina de Post. Nele, apresenta-se seu histórico; sua definição, com o auxílio de um exemplo explicativo; uma visão detalhada de uma de suas aplicações; uma breve discussão sobre sua importância, além da apresentação de um simulador de máquinas de Post desenvolvido pelas autoras.

Email addresses: luiza.engler@gmail.com (Luiza Engler Stadelhofer), carolinesala16@gmail.com (Caroline Sala de Borba)

2. Histórico

Post publicou, em 1936, um curto artigo no qual descrevia um modelo computacional muito similar ao modelo da máquina de Turing [3]. O artigo de Turing foi recebido para publicação no dia 28 de maio, enquanto o de Post foi recebido em 7 de outubro; uma nota de rodapé escrita por Alonzo Church, editor do artigo de Post, informa o leitor que por mais que o artigo possuía uma data mais antiga, foi escrito de forma completamente independente do de Turing [3].

Ao contrário de Turing, Post não usou em momento algum a palavra máquina para descrever seu modelo, mas sim a expressão "trabalhador ou solucionador de problemas" (contudo, o termo máquina de Post se tornou a terminologia padrão para o modelo) [3].

A funcionalidade análoga a fita da máquina de Turing foi providenciada por um "espaço de símbolo" que consistia em uma sequência infinita (em ambas as direções) de espaços ou caixas, na qual cada uma se encontra vazia/não marcada, ou marcada [4]. O trabalhador tem a capacidade de executar as seguintes ações [4]:

1. Marcar a caixa na qual ele está (supondo que esteja vazia);
2. Apagar a marcação da caixa na qual ele está (supondo que esteja marcada);
3. Mover para caixa à sua direita;
4. Mover para caixa à sua esquerda;
5. Determinar se a caixa na qual se encontra está marcada ou não.

O conjunto de direções dada ao trabalhador consiste em uma sequência numerada de instruções, cada uma tendo uma das seguintes formas [4]:

1. Realizar a operação 1, 2, 3 ou 4 e então seguir a direção j ;
2. Realizar a operação 5 e de acordo com a resposta seguir a direção número i (caso sim), e número j (caso não);
3. Parar.

Uma caixa deve ser destacada e denominada de ponto inicial. Um problema é dado de forma simbólica por um número finito de caixas marcadas, e a resposta é dada na mesma forma [4].

Foi desta maneira que Post inicialmente explicou seu modelo computacional. Entretanto, veremos a seguir a definição formal da conhecida máquina de Post, como assim é encontrada atualmente na literatura da área.

3. Definição

De acordo com [5] uma máquina de Post é composta por dois elementos principais:

- Uma variável X , que representa uma fila (FIFO – First In First Out) ilimitada, utilizada como entrada, saída e memória de trabalho da máquina;
- E um programa, que consiste em uma sequência de instruções representada através de um diagrama de fluxos (estrutura similar a um fluxograma).

3.1. Formalização

Uma máquina de Post MP é uma tripla [5]:

$$MP = (\Sigma, D, \#)$$

onde:

Σ – alfabeto de entrada;

D – diagrama de fluxos construído a partir das instruções básicas **partida**, **parada**, **desvio** e **atribuição**;

- símbolo auxiliar único.

As instruções básicas são definidas por [5] como a seguir:

- Partida - representa a instrução de início do diagrama. Só pode existir uma única partida por programa;
- Parada – representa a instrução de parada ou término do programa, podendo ser de dois tipos: aceitação ou rejeição. É possível existir várias instruções de parada em um único programa;
- Desvio ou Teste – desvio condicional que decide por qual caminho o fluxo do diagrama segue, de acordo com o símbolo armazenado na primeira posição da fila X. Deve ser feita a especificação de um desvio para cada letra do alfabeto de entrada, além de # e do símbolo vazio ϵ , como visto na figura 1. Se não existir a definição de um desvio de um certo símbolo para uma instrução, a máquina rejeita a entrada por indefinição, caso leia esse símbolo, estando na dita instrução. A operação $X \leftarrow \text{ler}(X)$ realiza uma leitura destrutiva da fila, ou seja, ao ser lido, o símbolo é retirado da mesma. Para facilitar o entendimento, esta operação é equivalente à operação *push* de uma *queue*, na linguagem C++, por exemplo.

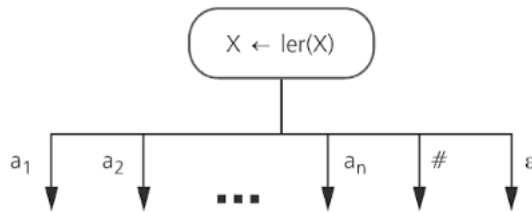


Figura 1: Exemplo de desvio condicional no diagrama de fluxos

Source: [5]

- Atribuição – faz a concatenação de um símbolo $s \in (\Sigma \cup \#)$ com a palavra armazenada na fila X, colocando o símbolo no final da fila, como observado na figura 2. Equivalente à operação *pop* de uma *queue* em C++.

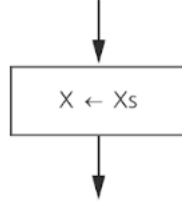


Figura 2: Exemplo de atribuição no diagrama de fluxos

Source: [5]

A linguagem aceita (ou rejeitada) por uma máquina de Post MP é o conjunto de palavras w tais que se a fila X é igual a w antes da execução do programa, então eventualmente a máquina de Post executa uma instrução de aceitação (ou rejeição, respectivamente) [6]. É importante observar que uma máquina de Post, assim como uma máquina de Turing, também pode entrar em loop, nunca alcançando uma instrução de aceitação ou rejeição [6].

4. Exemplo

Para mostrar com mais clareza o funcionamento de uma máquina de Post, vamos abordar um exemplo. Supondo uma linguagem livre de contexto L , tal que:

$$L = \{a^i b^j c^k \mid i + j = k \wedge i, j, k \geq 1\}$$

Podemos construir uma máquina de Post MP que decide L como a seguir:

$$MP = (\{a, b, c\}, D, \#)$$

Onde D é o diagrama de fluxos representado na figura 3.

Inicialmente, a fila X recebe a palavra de entrada, e.g. aabbbccccc (no caso de uma entrada a ser aceita) ou abbbcc (no caso de uma entrada a ser rejeitada). Em seguida, o símbolo auxiliar $\#$ é adicionado ao fim da fila - este servirá como controlador para identificarmos quando chegamos ao final de uma palavra. Ao ler da fila a primeira ocorrência do caractere a , não tomamos nenhuma ação, apenas retirando o mesmo da fila. Após isso, ao ler qualquer outro símbolo a ou b , iremos adicioná-los ao fim da fila. Quando encontrarmos a primeira ocorrência do símbolo c , apenas fazemos sua remoção, novamente sem executar nenhuma outra ação. A próxima instrução indica que ao ler qualquer outro c em sequência, devemos adicioná-lo a X . Lendo o símbolo $\#$, podemos concluir que chegamos ao fim da palavra, logo, podemos adicionar o caractere auxiliar mais uma vez a X e repetir o processo.

Quando a palavra de X não possuir mais nenhum a , a MP começa a fazer o mesmo procedimento com o símbolo b , retirando a cada passada um b e um c correspondente, ao invés de um a e um c , como nos primeiros passos.

A ideia geral usada para a construção dessa máquina de Post é cortar da palavra inicial, cada símbolo a ou b com um símbolo c associado, visto que queremos que $i + j$ (quantidade

de símbolos a mais quantidade de símbolos b) seja igual a k (quantidade de símbolos c). Neste exemplo, a instrução de rejeição foi omitida por questão de simplificação. Mas como dito anteriormente, qualquer desvio feito que não estiver especificado no diagrama de fluxos causará a máquina de Post a morrer/dar *crash*.

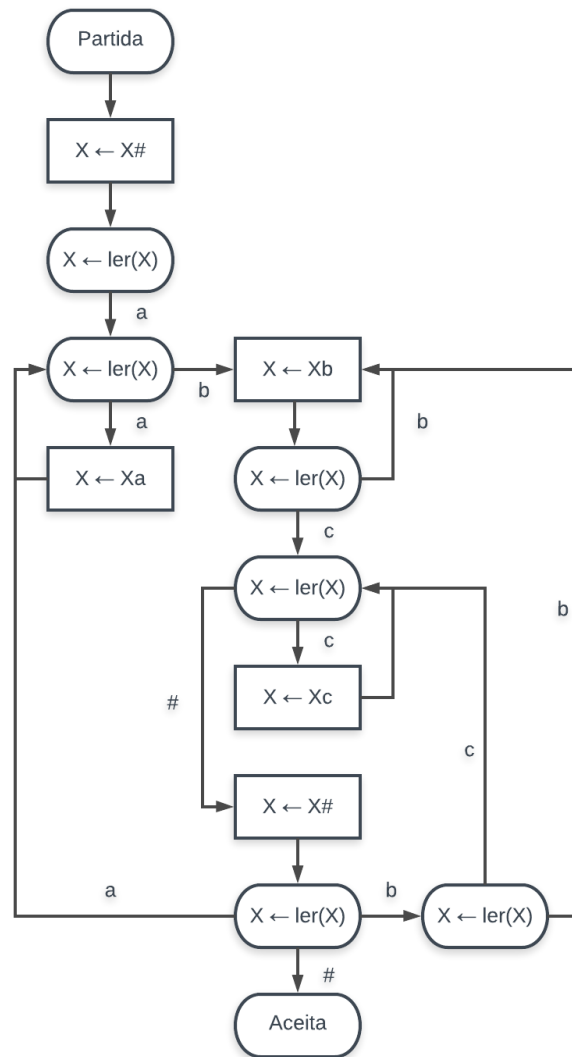


Figura 3: MP decisor de L
Source: Próprios Autores.

5. Aplicação: fundamentos da computação para crianças

Com a crescente evolução da tecnologia, o computador se tornou indispensável para as tarefas do dia a dia. Devido a onipresença dele em nossa sociedade, com aplicações em todos os níveis, muitas pessoas desconhecem o funcionamento de um computador e de um

programa [7]. A este respeito, retoma-se a importância da questão sobre familiarizar as crianças da escola com alguns conceitos básicos da computação [7]. Entretanto, alguns desafios estão relacionados a esse tema como o quão abstrato deve ser essa abordagem e qual a melhor metodologia para se utilizar com crianças, por exemplo.

Em 1979, o matemático russo Uspensky publicou um livreto chamado “Post’s Machine” na qual descreve como ele introduziu a noção abstrata de um programa de computador para as crianças em idade escolar[7]. No entanto, o autor não partiu de computadores reais e nem de linguagens de programação, mas de uma construção teórica, quase idêntica à de Turing, que é a máquina de post[7]. Essa abordagem contribui tanto para aprendizagem de fundamentos da computação, quanto para o treinamento do pensamento formal das crianças [7].

Uspensky primeiro descreve seus componentes e, em seguida, as instruções e possíveis classes de operações [7]. Depois desta descrição mais geral, ele discute mais alguns exemplos e exercícios [7]. Enquanto descreve o funcionamento da máquina, Uspensky não referencia aplicações concretas da máquina e nem diz que é uma máquina de computação abstrata, tendo em seu ponto vista a necessidade de aprendizagem de outros conceitos primeiramente [7].

O primeiro passo para ensinar o funcionamento da máquina é começar com uma descrição dos componentes e Uspensky usou terminologias diferentes de Post para os componentes, onde Post nomeou “espaço de símbolo”, “trabalhadores” e “caixas”, Uspensky nomeou como “fita”, “procedimento” e “cabeça de leitura ou gravação”[7]. Sempre que possível o autor do livreto destaca a importância da representação visual como metodologia de ensino e diz que a cada novo conceito introduzido é necessário uma visualização, por exemplo, sugindo o uso de uma fita de papel dividida em segmentos onde os objetos podem ser colocados manualmente[7]. Após a introdução dos componentes, pode-se começar a ilustrar os chamados atos primitivos de uma máquina de post: ela pode se mover para a esquerda ou para a direita; pode “examinar” e pode rotular ou apagar uma célula [7]. Todas essas ações devem ser novamente visualizadas através de exemplos [7].

O funcionamento da máquina não difere do funcionamento definido por Post, a máquina começa a partir da configuração de estado inicial, o cabeçote está voltado para uma das células, algumas das células da fita são rotuladas e outras não, e a máquina pode começar a executar um programa seguindo a primeira instrução na lista [7]. Existem três estados de finalização diferentes para uma máquina de post [7]: (1) A máquina entra em um estado não executável: ela precisa rotular uma célula já rotulada ou apagar uma célula não rotulada. Em seguida, a execução é interrompida e ocorre uma parada sem resultado. (2) A máquina chega a uma instrução de parada. (3) A máquina nunca para nem chega a uma instrução não executável: ela entra em um loop infinito.

A importância da exemplificação é sempre ressaltada. Exemplos adicionais de vários programas que levam aos tipos de finalização diferente (1), (2) e (3) são colocados, deixando claro dois aspectos: diferentes programas aplicados a mesma entrada podem levar a diferentes resultados e o mesmo programa aplicado a diferentes entradas pode também levar a resultados diferentes [7]. Características simples mas que são básicas para a programação.

Com os exercícios os alunos são estimulados a ir além com o conhecimento que possuem.

No primeiro exercício, as seguintes questões devem ser resolvidas: Pode haver um programa que sempre para, nunca interrompe ou resulta em uma instrução não executável, não importa qual seja a entrada? Qual é o tamanho mínimo para cada um desses programas? O segundo exercício pede um programa que possa realizar (1), (2) e (3) com diferentes entradas. Além disso, pede-se ao aluno que prove que o número mínimo de instruções para esse programa deve ser 4 [7]. Esses exercícios são importantes por várias razões, pois permitem um treinamento do pensamento formal e uma pequena análise do comprimento mínimo de programas que é tão fundamental na ciência da computação [7]. A criança é familiarizada com esses problemas sem estar ciente de seu significado prático e, portanto, possivelmente ganhando uma compreensão mais abstrata sobre este problema.

O autor do livreto escolheu a “adição unária” ou, também chamada, “cálculo de +1” como aplicação da máquina de post para discutir em todos os seus detalhes. Uma análise longa e detalhada desse exemplo simples é capaz de mostrar como ele pode não ser tão trivial como é considerado intuitivamente [7]. Segundo o autor do livreto, uma recomendação de metodologia de ensino para as crianças é não utilizar número e sim símbolos em todas as máquinas ilustradas [7]. Com a aplicação isso não foi diferente, Uspensky utiliza símbolos e não números para exemplificar o cálculo de +1 realizado, por exemplo, pode-se utilizar o símbolo V para representar o número 1 [7].

Detalhando a aplicação, cinco generalizações do estado de configuração inicial, que são crescentes em nível de complexidade, são mostradas juntamente com a lógica para encontrar um algoritmo para cada caso [7]. Todos os casos correspondem a posição do cabeçote em relação à entrada. Os casos são mostrados na figura 4.

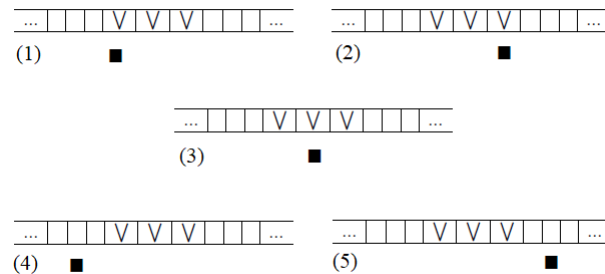


Figura 4: Possíveis casos do estado de configuração inicial

Source: [7, Adaptado]

Nos casos 1 e 2, o cabeçote é posicionado na célula marcada mais à esquerda ou mais à direita de uma sequência de células rotuladas que representa o número ao qual o 1 deve ser adicionado. No caso 3, o cabeçote pode estar em qualquer uma das células rotuladas. Nos casos 4 e 5, o cabeçote pode estar nas células em branco da esquerda ou da direita, aumentando a generalização e complexidade do algoritmo.

Na discussão dos casos 1 e 2, por exemplo, Uspensky prepara vários algoritmos para resolver o caso e como um exercício pede para provar que existem, na verdade, infinitos

programas para resolver este problema [7]. E segue com essa mesma metodologia para os outros casos, analisando e pedindo para que os alunos provem a quantidade de programas existentes ou o comprimento desses programas, reutilizem os passos de casos simples para resolver os casos complexos e até reduzir a quantidade de passos de um programa, semelhante a uma otimização [7].

Uspensky introduz uma maneira mais estruturada de escrever programas objetivando possibilitar a análise de programas longos sem perder a visão geral da solução, usando diagramas de blocos, semelhante à fluxogramas [7]. Hoje, é possível encontrar nas abordagens mais atuais a representação de um programa em uma máquina de Post por meio de fluxogramas, como em [5].

Após o detalhamento do algoritmo generalizado de adição unária, são considerados outros algoritmos que podem ser desenvolvidos com base no anterior, como por exemplo, algoritmos que adicionam um número arbitrário a um número já existente [7]. Com a mesma metodologia, a partir de um caso simples o problema é evoluído e se torna mais complexo a medida que fica mais geral.

Com um olhar geral, observa-se que, partindo de uma máquina muito simples, Uspensky mostrou como se pode de fato avançar os conceitos básicos de programação e computação para crianças em idade escolar [7]. No entanto, surge a dúvida, por que uma máquina abstrata, como uma máquina de Post, é o instrumento ideal para atingir esse objetivo? Para o autor do livreto, ao partir de uma máquina de Post sem fazer qualquer referência ao que é usado pode contribuir para um treinamento em pensamento formal, pois ao abstrair as aplicações as crianças podem ter uma concepção mais ampla da noção de computação [7].

Desta forma, esta formação em pensamento formal é prosseguida no ensino de algumas das aplicações de uma máquina de Post [7]. De fato, através da análise do caso de adição unária, a criança aprende que não há nada de fundamental na forma como se calcula e que tudo depende da estrutura formal da qual se está começando [7]. Entender que existem várias maneiras diferentes de fazer algo, pode tornar mais fácil abstrair os significados e interpretações específicos envolvidos ao trabalhar com um tipo específico de sistema e pode, assim, atribuir um treinamento em pensamento formal - uma habilidade básica para programação e computação [7].

6. Importância

Além do livreto de 1979 fruto do trabalho de Uspensky com as crianças em idade escolar e que tem como objetivo familiarizar as crianças com o pensamento formal utilizando a máquina de Post, encontrou-se outros trabalhos similares onde a máquina é utilizada como ferramenta de ensino.

No trabalho de [1], as máquinas de Turing e de Post são chamadas de “Ferramentas pedagógicas” e o autor menciona uma experiência onde o problema de representar o quociente entre dois inteiros foi apresentado a um grupo de estudantes do primeiro semestre, crianças, e solicitado que resolvessem com máquina de Post. Subentende-se que todo o embasamento sobre a máquina também lhes foi passado, mas ele não conheciam o procedimento da divisão

e de alguma forma, após uma semana, eles “re-descobriram” o algoritmo de divisão mais popular de hoje [1].

De maneira semelhante, o artigo [8] apresenta o ambiente que foi desenvolvido com base na máquina teórica de Post para utilizar na introdução aos métodos formais de programação. O autor justifica que embora a programação formal seja considerada um modelo de programação reconhecido, ela é evitada devido a seu alto custo mental.

Destacamos nesse trabalho a importância da máquina de Post como um ferramenta didática de ensino, seja de ensino de programação formal, conhecimento formal ou noções matemáticas. Entretanto, sabe-se que a definição de Emil Post e Alan Turing vieram no mesmo ano, de maneira independente, e que representaram um grande avanço para a computação. No artigo [6, pag. 44] há uma prova de equivalência entre a máquina de Post e de Turing, tal prova é devidamente extensa e portanto será omitida.

7. Simulador de Máquinas de Post

Com o intuito de fornecer uma maior contribuição para a área, apresentamos a seguir um simulador de máquinas de Post desenvolvido pelas autoras. O código fonte do sistema, implementado em C++, assim também como um modelo de arquivo de entrada - baseado no exemplo explicado na seção 4, e as instruções de execução, podem ser encontradas no seguinte repositório do GitHub: <https://bit.ly/2KhDim9>.

O simulador recebe como entrada um arquivo, que deve ser estruturado da seguinte maneira:

- Na primeira linha, deve ser informada a quantidade de estados que a máquina de Post possui, ou seja, o número de classes que representam o diagrama de fluxos;
- Em sequência, uma lista especificando o nome de cada um dos estados deve ser apresentada. É obrigatório que cada estado possua um identificador único, e que, no caso de ser um estado do tipo desvio, tenha um identificador que inicie com a palavra-chave *READ*, ou se for do tipo atribuição, inicie com a palavra *ADD*;
- Na terceira linha, temos o nome do estado de partida;
- E na quarta, temos o nome do estado de aceitação. É importante esclarecer que para simplicidade, optamos por permitir a existência de apenas um estado de aceitação, em contrapartida a uma máquina de Post, que pode conter vários. Além disso, o estado de rejeição foi omitido no sistema, de forma que a máquina de Post simulada rejeita as entradas apenas por indefinição;
- Em seguida, temos um número n que representa a quantidade de transições ou ligações existentes no diagrama de fluxos;
- Nas n linhas seguintes, devem ser especificadas cada uma das ligações, de modo que após um estado *ADD*, seja escrito o símbolo que queremos adicionar à fila; e após um estado *READ*, seja apresentado um símbolo ao qual quando lido da fila, leve ao estado definido na sequência;

- Por último, na linha final do arquivo especificamos qual palavra de entrada queremos testar no simulador.

No simulador, o diagrama de fluxos da máquina de Post foi representado através de um grafo, sendo armazenado em uma lista de adjacências. A simulação da computação da palavra de entrada foi feita utilizando uma estrutura auxiliar do tipo fila, do mesmo modo como a máquina de Post utiliza uma fila para lidar com a entrada, a saída e a memória de trabalho.

8. Conclusão

Nesse artigo foi possível descrever um breve histórico da máquina teórica de Post, suas particularidades e algumas diferenças com relação a de Turing. Emil Post ao definir as tarefas que os trabalhadores poderiam realizar como sendo marcar, apagar ou examinar a caixa atual, mover para a caixa da direita ou esquerda, com um conjunto de caixas infinitas, forneceu a base teórica para um importante dispositivo de computação.

Uma visão detalhada de uma aplicação da máquina de Post para ensinar e familiarizar crianças em idade escolar com conceitos básicos de computação pôde revelar a importância da máquina de Post como ferramenta pedagógica para ensinar a teoria e o funcionamento de um computador, que por vezes fica adjacente as interfaces gráficas em computadores reais. A utilização da máquina de Post como tal ferramenta teve o intuito de abstrair a realidade dando as crianças uma concepção mais ampla da realidade, além de permitir que elas visualizem o passo a passo de uma solução e as diversas soluções existentes para um mesmo problema.

Por fim, simulador criado permitiu aplicar os conhecimentos adquiridos e contribuir para a área de pesquisa em questão. Escrito em C++ ele simula a máquina de Post através de um arquivo de entrada que contém as especificações em um formato pré-definido.

Referências

- [1] N. Urrego, Some applications of post and turing machines in mathematics teaching, in: Proceedings of the Fourth International Derive TI-89/92 Conference, Liverpool, UK, 2000.
- [2] D. I. A. Cohen, Introduction to Computer Theory, John Wiley & Sons, Inc., New York, USA, 1986.
- [3] A. Urquhart, Emil post, Handbook of the History of Logic 5 (2009) 617–666. doi:[https://doi.org/10.1016/S1874-5857\(09\)70016-0](https://doi.org/10.1016/S1874-5857(09)70016-0).
- [4] E. L. Post, Finite combinatory processes - formulation 1, The Journal of Symbolic Logic 1 (3) (1936) 103–105. doi:<https://doi.org/10.2307/2269031>.
- [5] T. A. Diverio, P. B. Menezes, Teoria da computação: máquinas universais e computabilidade, Sagra Luzzatto, Porto Alegre, BR, 1999.
- [6] A. Pettorossi, Elements of Computability, Decidability, and Complexity, Aracne, Rome, IT, 2014.
- [7] L. de Mol, Post's machine. doi:<http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.72.6205>.
- [8] V. Dagdilelis, M. Satratzemi, Post's machine: A didactic microworld as an introduction to formal programming, Education and Information Technologies 6 (2) (2001) 123–141. doi:10.1023/A:1012319900191.
URL <https://doi.org/10.1023/A:1012319900191>